Lab week 3-1: Searching in Arrays

Let M be an mxn matrix of integers in which the integers in every row are in increasing order from smallest row index to largest row index and the integers in every column is in increasing order from smallest column index to largest column index. See below for an example.

Design and efficient algorithm (e.g. O(n)) that finds a given integer or concludes that the integer is not in the array. Your method should return an array of two numbers. The row and column where the number is found or [-1,-1] if the number is not found in the array.

Example

```
int[][] board1 = new int[][] {
{1, 2, 8, 9},
{3, 6, 12, 13},
{7,10,13, 29},
{10, 11, 28, 30}
};
int target = 12;
```

The call **search(board1, target)**   **will return {1,2} since 12 is in row 1 and column 2 (indexing as usual starts at 0.**

Submission

- Implement and submit to PolyLearn a **public class ArraySearch** that has a single public static method with the following signature.
- **public static int[] search (int[][] currArray, int target)**
  - currArray is the two dimensional array of integers to be searched
  - target is the integer to be found